Simulation of the Collatz 3x+1 function by Turing machines

Pascal Michel

Simulation of the Collatz $3x + 1$ function by Turing machines

Pascal MICHEL[*]
Équipe de Logique Mathématique,
Institut de Mathématiques de Jussieu – Paris Rive Gauche, UMR 7586,
Bâtiment Sophie Germain, case 7012, 75205 Paris Cedex 13, France
and Université de Cergy-Pontoise, IE, F-95000 Cergy-Pontoise, France
michel@math.univ-paris-diderot.fr

September 24, 2014

Abstract

We give new Turing machines that simulate the iteration of the Collatz $3x + 1$ function. First, a never halting Turing machine with 3 states and 4 symbols, improving the known 3×5 and 4×4 Turing machines. Second, Turing machines that halt on the final loop, in the classes 3×10, 4×6, 5×4 and 13×2.

Keywords: Collatz function, $3x + 1$ function, Turing machines.

Mathematics Subject Classification (2010): *Primary* 03D10, *Secondary* 68Q05, 11B83.

1 Introduction

Turing machines can be classified according to their numbers of states and symbols. It is known (see [8] for a survey) that there are universal Turing machines in the following sets (number of states $\times$ number of symbols):

$$2 \times 18,\ 3 \times 9,\ 4 \times 6,\ 5 \times 5,\ 6 \times 4,\ 9 \times 3,\ 18 \times 2.$$

On the other hand, all the Turing machines in the following sets are decidable:

$$1 \times n,\ 2 \times 3,\ 3 \times 2, n \times 1.$$

In order to refine the classification of Turing machines between universal and decidable classes, properties in connection with the $3x + 1$ function have been considered.

*Corresponding address: 59 rue du Cardinal Lemoine, 75005 Paris, France.

Recall that the $3x+1$ function T is defined by

$$T(x) = \begin{cases} x/2 & \text{if } x \text{ is even} \\ (3x+1)/2 & \text{if } x \text{ is odd} \end{cases}$$

This can also be written $T(2n) = n$, $T(2n+1) = 3n+2$. When function T is iterated on a positive integer, it seems that the loop $2 \mapsto 1 \mapsto 2$ is always reached, but this is unproven, and is a famous open problem in mathematics [3]. For further references, we set

> **$3x+1$ Conjecture**: When function T is iterated from positive integers, the loop $2 \mapsto 1 \mapsto 2$ is always reached.

The $3x+1$ function is also called the Collatz function, and *Collatz-like* functions are functions on integers with a definition of the following form: there exist integers $d \geq 2$, a_i, b_i, $0 \leq i \leq d-1$, such that, for all integers x,

$$f(x) = \frac{a_i x + b_i}{d} \quad \text{if} \quad x \equiv i \pmod{d}.$$

With these definitions, we can state the following properties of Turing machines, that have been used to refine the classification according to the numbers of states and symbols (see [7] for a survey).

- Turing machines that simulate the iteration of the $3x+1$ function and never halt. It is known that there are such machines in the sets

  $$2 \times 8,\ 3 \times 5,\ 4 \times 4,\ 5 \times 3,\ 10 \times 2.$$

  We improve these results by giving a 3×4 Turing machine.

- Turing machines that simulate the iteration of the $3x+1$ function and halt when the loop $2 \mapsto 1 \mapsto 2$ is reached. It is known that there is such a machine in the set 6×3. In this article, we give four new Turing machines, in the classes 3×10, 4×6, 5×4 and 13×2.

- Turing machine that simulate the iteration of a Collatz-like function. It is known that there are such machines in the sets

  $$2 \times 4,\ 3 \times 3,\ 5 \times 2.$$

2 Preliminaries: Turing machines

The Turing machines we use have

- one tape, infinite on both sides, made of cells containing symbols,
- one reading and writing head,
- a set $Q = \{A, B, \ldots\}$ of states, plus a halting state H (or Z),

symbols													
10	Ma	**Mi**$_2$											
9													
8	Ba												
7													
6		Ma	**Mi**$_2$										
5		Ba											
4		Mi_2	Ma	**Mi**$_2$									
3				Ma	**Mi**$_1$								
2									Ba	Ma		**Mi**$_2$	
	2	3	4	5	6	7	8	9	10	11	12	13	states

Table 1: Turing machines simulating the $3x + 1$ function: Ma = Margenstern [4, 5], Ba = Baiocchi [1], Mi_1 = Michel [6], Mi_2 = Michel (this paper). In roman boldface, halting machines.

- a set $\Sigma = \{b, 0, 1, \ldots\}$ of symbols, where b is the blank symbol (or $\Sigma = \{0, 1\}$, when 0 is the blank symbol),
- a next move function

$$\delta : Q \times \Sigma \to \Sigma \times \{L, R\} \times (Q \cup \{H\}).$$

If $\delta(p, a) = (b, D, q)$, then the Turing machine, reading symbol a in state p, replaces a by b, moves in the direction $D \in \{L, R\}$ (L for Left, R for Right), and comes into state q. On an input $x_k \ldots x_0 \in \Sigma^{k+1}$, the initial configuration is ${}^{\omega}b(Ax_k) \ldots x_0 b^{\omega}$. This means that the word $x_k \ldots x_0$ is written on the tape between two infinite strings of blank symbols, and the machine is reading symbol x_k in state A.

3 The known Turing machines

Let us give some more precisions about the Turing machines that simulate the $3x+1$ function. The following results are displayed in Table 1.

Michel [6] gave a 6×4 Turing machine that halts when number 1 is reached. This machine works on numbers written in binary. Division by 2 of even integers is easy and multiplication by 3 is done by the usual multiplication algorithm.

Margenstern [4, 5] gave never halting 5×3 and 11×2 Turing machines in binary, and never halting 2×10, 3×6, 4×4 Turing machines in unary, that is working on numbers n written as strings of n 1s.

Baiocchi [1] gave five never halting Turing machines in unary, including 2×8, 3×5 and 10×2 machines that improved Margenstern's results.

In this article, we give a never halting 3×4 Turing machine that works on numbers written in base 3. Multiplication by 3 is easy and division by 2 is done by the usual division

algorithm. Note that Baiocchi and Margenstern [2] already used numbers written in base 3 to define cellular automata that simulate the $3x+1$ function.

By adding two states to this 3×4 Turing machine, we derive a 5×4 Turing machine that halts when number 1 is reached.

We also give three other Turing machines that halt when number 1 is reached:

- A 3×10 Turing machine obtained by adding one state to the 2×10 Turing machine of Margenstern [4, 5].
- A 4×6 Turing machine obtained by adding one state to the 3×6 Turing machine of Margenstern [4, 5].
- A 13×2 Turing machine obtained by adding two states to the 11×2 Turing machine of Margenstern [4, 5].

4 A never halting 3×4 Turing machine

This Turing machine M_1 is defined as follows:

M_1	b	0	1	2
A	bLC	0RA	0RB	1RA
B	2LC	1RB	2RA	2RB
C	bRA	0LC	1LC	2LC

The idea is simple. A positive integer is written on the tape, in base 3, in the usual order. Initially, in state A, the head reads the most significant digit, at the left end of the number. The initial configuration on input $x=\sum_{i=0}^{k} x_i 3^i$ is ${}^{\omega}b(Ax_k)\ldots x_0 b^{\omega}$. Then the machine performs the division by 2, using the usual division algorithm. Partial quotients are written on the tape. Partial remainders are stored in the states: 0 in state A, 1 in state B. When the head passes the right end of the number, reading a b, then

- if the remainder is 0, nothing is done: $2n\mapsto n$,
- if the remainder is 1, a 2 is concatenated to the number: $2n+1\mapsto n\mapsto 3n+2$.

Then the head comes back, in state C, to the left end of the number and is ready to perform a new division by 2.

We have the following theorem.

Theorem 4.1 *The $3x+1$ conjecture is true iff, for all positive integer $x=x_k\ldots x_0$ written in base 3, there exists an integer $n\geq 0$ such that, on input $x_k\ldots x_0$, the Turing machine M_1 eventually reaches the configuration ${}^{\omega}b0^n(A1)b^{\omega}$.*

5 Turing machines that halts on the final loop

5.1 A 3×10 Turing machine

Margenstern [5, Fig. 11] gave the folowing never halting 2×10 Turing machine M_2.

M_2	b	1	x	r	u	v	y	z	t	k
A	bRA	xRB	1LA	kRB	xRA	xRA	rLA	rLA	yRA	
B	zLB	uRB	xRB	yRB	vLB	uRA	tLB	1LA	xRB	bRB

Turing machine M_2 works on numbers written in unary, so that the initial configuration on number $n \geq 1$ is ${}^{\omega}b(A1)1^{n-1}b^{\omega}$. By adding a new state C, we can detect the partial configuration $(A1)b$, and we obtain the following 3×10 Turing machine M_3.

M_3	b	1	x	r	u	v	y	z	t	k
A	bRA	xRC	1LA	kRB	xRA	xRA	rLA	rLA	yRA	
B	zLB	uRB	xRB	yRB	vLB	uRA	tLB	1LA	xRB	bRB
C	bLH	uRB		yRB						

We have the following theorem

Theorem 5.1 *The $3x+1$ conjecture is true iff, for all positive integers n, Turing machine M_3 halts on the initial configuration ${}^{\omega}b(A1)1^{n-1}b^{\omega}$.*

5.2 A 4×6 Turing machine

Margenstern [5, Fig. 10] gave the folowing never halting 3×6 Turing machine M_4 (Note that transition $(1, z) \mapsto (x\text{R}2)$ in this figure should be $(1, z) \mapsto (r\text{R}2)$).

M_4	b	1	x	a	z	r
A	bRA	xRB	1LA	1LA	rRB	
B	1LB	aRC	1LB	1LA	xRB	bRA
C	zLC	xRC	1LC	aRA	rRC	zLC

Turing machine M_4 works on numbers written in unary, with initial configuration ${}^{\omega}b(A1)1^{n-1}b^{\omega}$. By adding a new state D, we can detect the partial configuration $(A1)b$, and we obtain the following 4×6 Turing machine M_5.

M_5	b	1	x	a	z	r
A	bRA	xRD	1LA	1LA	rRB	
B	1LB	aRC	1LB	1LA	xRB	bRA
C	zLC	xRC	1LC	aRA	rRC	zLC
D	bLH	aRC			xRB	

We have the following theorem.

Theorem 5.2 *The $3x+1$ conjecture is true iff, for all positive integers n, Turing machine M_5 halts on the initial configuration ${}^{\omega}b(A1)1^{n-1}b^{\omega}$.*

5.3 A 5×4 Turing machine

This Turing machine M_6 is defined as follows.

M_6	b	0	1	2
A	bLC	0RA	0RB	1RA
B	2LE	1RB	2RA	2RB
C	bRD	0LC	1LC	2LC
D		bRA	bRB	1RA
E	bRH	0LC	1LC	2LC

Turing machine M_6 is obtained from Turing machine M_1 by adding a state D that wipes out the useless 0s, and a state E that detects the partial configuration $b(Bb)$.

We have the following theorem.

Theorem 5.3 *The $3x + 1$ conjecture is true iff Turing machine M_6 halts on all input $x = x_k \dots x_0$ representing a positive integer written in base 3.*

5.4 A 13×2 Turing machine

Margenstern [5, Fig. 8] gave the following never halting 11×2 Turing machine M_7 (in this table, H is *not* a halting state).

M_7	0	1
A	1RI	0RB
B	0RA	0RG
C	0RA	1RD
D	0RC	1RE
E	1RI	1RF
F	1RC	0RG
G	1RC	1RH
H	0RE	1RG
I	1LJ	
J	0RB	1LK
K	0LJ	1LJ

This machine works on numbers written in binary, with the least significant bit at the left end of the number, and digits 0 and 1 coded by 10 and 11, so that the initial configuration on number $n = x_k \dots x_0 = \sum_{i=0}^{k} x_i 2^i$ is ${}^{\omega}0(A1)x_0 1x_1 \dots 1x_k 0^{\omega}$. Division by 2 of even integers is easy, and multiplication by 3 is done by the usual algorithm.

By adding two new states L and M, we can detect the partial configuration $(A1)10$, and we obtain the following 13×2 Turing machine M_8, where Z is the halting state.

M_8	0	1
A	1RI	0RL
B	0RA	0RG
C	0RA	1RD
D	0RC	1RE
E	1RI	1RF
F	1RC	0RG
G	1RC	1RH
H	0RE	1RG
I	1LJ	
J	0RB	1LK
K	0LJ	1LJ
L	0RA	0RM
M	0LZ	1RH

We have the following theorem.

Theorem 5.4 *The $3x+1$ conjecture is true iff, for all positive number $n = x_k \dots x_0 = \sum_{i=0}^{k} x_i 2^i$, Turing machine M_8 halts on the initial configuration ${}^{\omega}0(A1)x_0 1 x_1 \dots 1 x_k 0^{\omega}$.*

6 Conclusion

We have given a new 3×4 never halting Turing machine that simulates the iteration of the $3x+1$ function. It seems that it will be hard to improve the known results on never halting machines.

On the other hand, for Turing machines that halt on the conjectured final loop of the $3x+1$ function, more researches are still to be done.

References

[1] C. Baiocchi, 3N+1, UTM e Tag-systems (Italian), Dipartimento di Matematica dell'Università "La Sapienza" di Roma **98/38**, 1998.

[2] C. Baiocchi and M. Margenstern, Cellular automata about the $3x+1$ problem, in: Proc. LCCS'2001, Université Paris 12, 2001, 37–45, available on the website http://lacl.univ-paris12.fr/LCCS2001/.

[3] J.C. Lagarias (Ed.), The Ultimate Challenge: The $3x+1$ Problem, AMS, 2010.

[4] M. Margenstern, Frontier between decidability and undecidability: a survey, Proc. MCU'98, Vol. 1, ISBN 2-9511539-2-9, 1998, 141–177.

[5] M. Margenstern, Frontier between decidability and undecidability: a survey, *Theoret. Comput. Sci.* **231**, 2000, 217–251.

[6] P. Michel, Busy beaver competition and Collatz-like problems, *Arch. Math. Logic* **32** (5), 1993, 351–367.

[7] P. Michel and M. Margenstern, Generalized $3x+1$ functions and the theory of computation, in [3], 105–128.

[8] D. Woods and T. Neary, The complexity of small universal Turing machines: a survey, *Theoret. Comput. Sci.* **410**, 2009, 443–450. Extended and updated in: http://arxiv/abs/1110.2230.